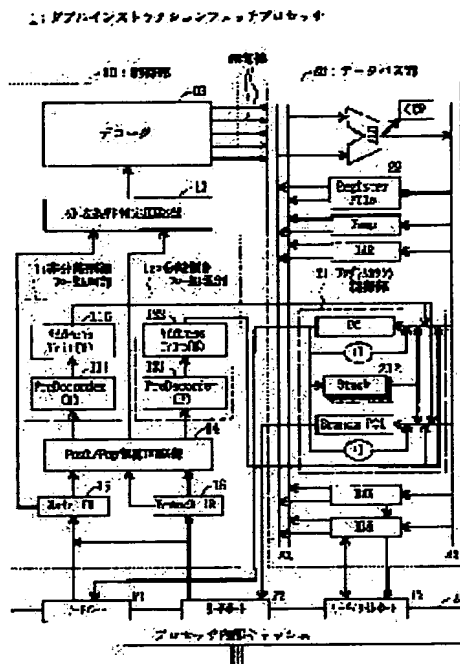


(11)Publication number : 09-282163  
(43)Date of publication of application : 31.10.1997

(72)Inventor : ISHII KENJI  
TSUTSUI AKIHIRO  
MIYAZAKI TOSHIAKI

**SOLUTION:** This processor has a control part 10, data path part 20 and internal processor cache 30. Then, the address of the branch destination is extracted from a branching instruction fetched on an instruction fetch stage, the instruction code corresponding to both the control flow in the case of not establishing the branch and the control flow in the case of establishing the branch is fetched on the instruction stage of the next clock, the instruction code is selected corresponding to the presence/absence of branch based on the executed result of the preceding test instruction at an ALU and the branching conditions, and this selected instruction code is inputted to the instruction decode stage.



[Date of extinction of right]

BEST AVAILABLE COPY

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-282163

(43) 公開日 平成9年(1997)10月31日

(51) Int.Cl.<sup>4</sup>

G06F 9/38

識別記号

.930

片内整理番号

PI

G06F 9/38

技術表示箇所

330F

審査請求 未請求 請求項の数 2 FD (全 11 頁)

(21) 出願番号 特願平8-118511

(22) 出願日 平成8年(1996)4月16日

(71) 出願人 00004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 石井 健司

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(72) 発明者 筒井 章博

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(72) 発明者 宮崎 敏明

東京都新宿区西新宿三丁目19番2号 日本

電信電話株式会社内

(74) 代理人 弁理士 川久保 新一

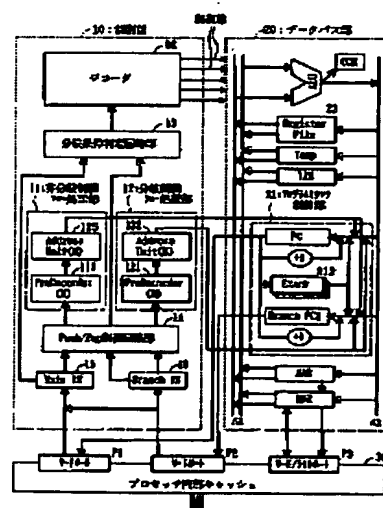
(54) 【発明の名称】 ダブルインストラクションフェッチプロセッサ

(57) 【要約】

【課題】 バイブライン化されたプロセッサにおいて、小規模のハードウェアによって、常に正常な制御フローを保證することができ、制御ハザードによるバイブラインストールが発生しないプロセッサを提供することを目的とするものである。

【解決手段】 n段バイブライン化された各種プログラム処理方式のプロセッサにおいて、インストラクションフェッチステージでフェッチした分枝命令から分枝先アドレスを取り出し、分枝不成立の場合と分枝成立の場合との両方の制御フローに対応する命令コードを、次のクロックのインストラクションステージでフェッチし、その後、先行するテスト命令の実行結果と分枝条件とに基づいて、分枝の有無に応じて決定された命令コードを選択し、インストラクションデコードステージに投入するものである。

1. ダブルインストラクションフェッチプロセッサ



32700

(2)

特開平9-282163

1

## 【特許請求の範囲】

【請求項1】 n段パイプライン化された書翰プログラム処理方式のプロセッサにおいて、  
 インストラクションフェッチステージでフェッチした分岐命令から分岐先アドレスを取り出し、分岐不成立の場合の制御フローと分岐成立の場合の制御フローとの両方の制御フローに対応する命令コードを、次のクロックのインストラクションステージでフェッチし、先行するテスト命令の実行結果と分岐条件とに基づいて、分岐の有無に応じて決定された命令コードを選択し、この選択された命令コードをインストラクションデコードステージに投入することを特徴とするダブルインストラクションフェッチプロセッサ。

【請求項2】 n段パイプライン化された書翰プログラム処理方式のプロセッサにおいて、  
 分岐情報を含む命令コードを命令フェッチステージで検出し、この検出された命令コードの中から、上記分岐情報を取り出すプリデコード回路と；上記プリデコード回路で取り出された上記分岐情報に基づいて、分岐先アドレスを決定するアドレス計算回路と；上記アドレス計算回路で決定されたアドレスにアクセスし、分岐先の命令コードをフェッチし、このフェッチされた分岐先の命令コードを格納するブランチ用インストラクションレジスタと；分岐成立または分岐不成立を決める分岐条件についての判定結果に応じて、通常のインストラクションレジスタの出力と上記ブランチ用インストラクションレジスタの出力とのうちの一方の出力を選択し、この選択された出力を命令デコードステージに送り込むインストラクションレジスタ選択手段と；を有し、制御ハザードに起因するパイプラインストールを発生させずに制御フローの変更を行うことを特徴とするダブルインストラクションフェッチプロセッサ。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、書翰プログラム処理方式におけるプロセッサの構成に関するものである。

【0002】

【従来の技術】書翰プログラム処理方式とは、処理の手順を記述した一連の命令コードをメモリの記憶素子に書き、この書き込まれた命令コードをプロセッサが逐一読み出し、この読み出された命令コードの指示に従って、処理を1ステップづつ進めてゆく方式である。プロセッサが提供する演算は汎用的な関数であるので、1命令コードで処理できる演算は限られたものである。したがって、所望の機能を実現するために複雑な演算を必要とする場合には、単純計算を多数組み合わせる必要があり、多量の命令コードとクロック数とを消費する。

【0003】通常、1命令コードについての処理は、次のような①～⑤のステージを経て完了する。

① インストラクション（命令）フェッチ・ステージ

（IFステージ）

次に実行するプログラム（命令コード）を読み出すステージである。

② インストラクション（命令）デコード・ステージ（IDステージ）

読み出した命令コードを解読し、演算を行うデータバスの状態を設定するステージである。

③ エグゼキューション（実行）・ステージ（EXステージ）

10 内部レジスタに格納されたデータを処理対象とし、算術論理演算ユニットによって各種の演算を行う。または、メモリアクセスのためのアドレスの計算を行うステージである。

④ メモリアクセス・ステージ（MEMステージ）

メモリにデータのロードまたはストアを行うステージである。

⑤ ライトバック・ステージ（WBステージ）

計算結果または、ロードされたデータを内部レジスタに書き戻すステージである。

20 【0004】ここで、「パイプライン化」とは、上記各ステージ毎、または、さらに細かい単位で処理を分割し、これら分割された処理を並列動作させる技術である。つまり、1クロック毎にラッチ等で区切られた各ステージを、命令コードが順々に通過し、また、1クロック毎に、新しい命令コードがパイプラインに投入される。したがって、パイプライン化されたプロセッサにおいては、複数の命令コードが異なるステージで同時に処理される。たとえば、命令コードC1をWBステージで処理しているときに、命令コードC2がMEMステージで処理され、命令コードC3がEXステージで処理され、命令コードC4がIDステージで処理され、命令コードC5がIFステージで処理され、これらの処理が同時に実行される。なお、分割されたステージの総数を、パイプライン段数と呼ぶ。

【0005】命令コードの処理は、全てのステージを経て完了するので、1命令コードの処理に要する時間自体は、パイプライン化によって減少するわけではない。しかし、1クロック毎に、1命令コードの処理が完了するので、処理のスループットが向上する。これは、高度なパイプライン化によって、1ステージ当りの処理量が減少し、1クロック時間を小さくすることができるからである。

【0006】最近の汎用プロセッサで盛んに用いられているスーパーパイプライン技術は、上記パイプライン段数を増やしたものであり、より細かく処理を分割することによって、1クロックの時間をさらに小さくし、スループットを一層向上させることを狙ったものである。

【0007】

【発明が解決しようとする課題】パイプラインは、書翰プログラム処理方式における処理速度の向上に極めて有

50

(3)

特開平9-282163

効な技術であるが、しかし、処理の順序関係が予め決められている複数の命令コードを同時に処理するので、ハードウェアリソースが競合したり、処理中の命令コードが矛盾する等、パイプラインの正常な処理が乱される場合がある。これをパイプラインハザードと呼ぶ。

【0008】数あるパイプラインハザードの1つに、制御ハザードがある。「制御ハザード」とは、条件分岐や無条件ジャンプのために制御フローが変更されることによって生じるパイプライン中の命令コードの矛盾である。ここで、条件分岐とは、ある条件（たとえば、ある特定のレジスタの状態）によって、制御フローの変更の有無が変化する命令であり、無条件ジャンプとは、制御フローの変更を必ず行う命令である。以下では、特に言明しない限り、条件分岐と無条件ジャンプを区別せず、どちらも分岐と呼ぶ。したがって、この場合、無条件ジャンプの分岐条件は常に真（分岐成立）である。

【0009】命令コードをフェッチする場合、通常、単調増加するプログラムカウンタの値をアドレスにするので、たとえば、アドレス $n$ の命令コードがEXステージで処理された結果、分岐が確定したとし、この分岐確定によって、プログラムカウンタの値が $m$ に変更された場合、IDステージには、アドレス $n+1$ の命令コードが既に投入され、IFステージには、アドレス $n+2$ の命令コードが既に投入されている。しかし、上記分岐が確定された後には、そのパイプライン中には、本来、アドレス $m$ 、 $m+1$ の命令コードがそれぞれ投入されるべきであり、したがって、パイプライン中に実際に投入されている命令コードと、分岐確定後における正常な命令順序との間に矛盾が生じている。

【0010】制御ハザードによってパイプラインハザードが発生すると、パイプラインの矛盾状態を解消するために、パイプラインはストールする。すなわち、処理を完了していない全ての命令コードを破棄し、間違った命令コードが投入される直前のパイプラインの状態まで戻す。制御ハザードによって上記パイプラインストールが発生すると、高速処理にとって、極めて大きなペナルティになるので、このパイプラインストールの頻度を低減または解消する方法が考案されている。たとえば、プログラムの改良によって分岐の絶対数を減らしたり、分岐命令のスケジューリングといったソフトウェア的な方法を採用する他に、分岐予測や投機的実行といったハードウェア的なサポート機構を用いる。

【0011】「分岐予測」は、過去の履歴や統計情報を使って制御フローの変更を予測するものであり、制御回路の他に、過去の分岐のログを記録するためのバッファメモリ等、新たなハードウェアが必要になる。また、分岐予測を行っても、その予測が外れた場合には、パイプラインストールが発生する。

【0012】「投機的実行」は、複数の制御フロー上の命令コードをインターリーブして処理する動作である。

分岐の条件を毎クロックチェックし、分岐が確定した時点で、誤りがあった制御フロー上の命令コードの処理結果を無効化する。この場合、パイプラインストールは発生しないが、投機的実行中の実行処理速度が低下するという問題がある。また、複雑な制御回路を必要とするという問題がある。

【0013】本発明は、パイプライン化されたプロセッサにおいて、小規模のハードウェアによって、常に正常な制御フローを保證することができ、制御ハザードによるパイプラインストールが発生しないプロセッサを提供することを目的とするものである。

【0014】

【課題を解決するための手段】本発明は、 $n$ 段パイプライン化された蓄積プログラム処理方式のプロセッサにおいて、インストラクションフェッチステージでフェッチした分岐命令から分岐先アドレスを取り出し、分岐不成立の場合と分岐成立の場合との両方の制御フローに対応する命令コードを、次のクロックのインストラクションステージでフェッチし、その後、先行するテスト命令の実行結果と分岐条件とに基づいて、分岐の有無に応じて決定された命令コードを選択し、インストラクションデコードステージに投入するものである。

【0015】

【発明の実施の形態および実施例】図1は、本発明の一実施例である蓄積プログラム処理方式のダブルインストラクションフェッチプロセッサ1を示すブロック図である。

【0016】ダブルインストラクションフェッチプロセッサ1は、制御部10と、データバス部20と、プロセッサ内部キャッシュ30とを有する。

【0017】制御部10は、非分岐制御フロー処理部11と、分岐制御フロー処理部12と、分岐条件判定回路部13と、Push/Pop制御回路部14と、デコードDEと、Main IR15と、Branch IR16とを有する。

【0018】データバス部20は、プログラムカウンタ制御部21と、Register File22とを有する。

【0019】プロセッサ内部キャッシュ30は、リードポートP1、P2と、リード/ライトポートP3とを有する。

【0020】非分岐制御フロー処理部11は、特定の場台（分岐が成立し、Branch用プログラムカウンタBranch PC1のアドレスで読み出した命令コードをフォワードする場合等）を除いて、プログラムカウンタPCによって示されるアドレスから読み出した命令コードに応じて、主に、分岐命令の検出と分岐先アドレスの計算とを行う部分である。

【0021】通常、プログラムカウンタPCは、それまで実行していた命令コードに連続するアドレスや、無条

(4)

特開平9-282163

5

件ジャンプや、スタックのポップによって設定されたアドレスを保持しており、正常な制御フローのアドレスを管理している。しかし、バイライン化されたプロセッサにおいて、条件分岐命令が検出された状態で、保持されているアドレスは、分岐不成立の場合の制御フローとみなせる。

【0022】図2は、ダブルインストラクションフェッチプロセッサ1における制御部10を、プロセッサ内部キャッシュ30とともに示す図である。

【0023】図3は、ダブルインストラクションフェッチプロセッサ1におけるデータバス部20を、プロセッサ内部キャッシュ30とともに示す図である。

【0024】リードポートP1を介して、プログラムカウンタPCから出力されたアドレスに応じて読み出された命令コードは、Main IR（インストラクションレジスタ）15に入力される。直ちに非分岐制御フロー処理部11のPre Decoder（M）111が、Main IR15で保持された命令コードから、制御フローを変化させる分岐命令（条件分岐/無条件ジャンプ/Call/Return）を検出し、分岐条件112と分岐先アドレス情報とを抽出する。

【0025】この抽出された分岐条件112は分岐条件判定回路131に転送される。また、分岐先アドレス情報からプロセッサ1のアドレッシングモードを判定し、即値113やアドレス計算に関する情報114をAddress Unit（M）115に転送する。これとともに、レジスタ間接アドレスの場合や、レジスタの値をベースアドレスとする相対アドレス指定等の場合は、Register File22へRegister Select（M）信号を送出し、汎用レジスタからAddress Unit（M）115へデータの転送を指示する。

【0026】Pre Decoder（M）111によるブリデコードによって、CallまたはReturn命令が検出された場合は、Push/Pop制御回路部14に対して、それぞれ、Call Op.（M）信号141、Ret Op.（M）信号142を送出する。また、条件分岐または無条件ジャンプが検出された場合、データバス部20の特定のセレクトを切り替え、分岐先アドレスをフォワードする。すなわち、無条件ジャンプ命令が検出された場合は、制御フローの変更がこの時点で確定しているため、データバス部20のセレクトSel. 4に、無条件Jump Op.（M）信号を送出し、プログラムカウンタPCの出力をJump Address（M）に切り替える。ただし、上記実施例では、Return命令も一種の無条件ジャンプとして検出することとする。

【0027】Pre Decoder（M）111がブリデコードし、分岐条件判定回路131が条件分岐命令を検出した場合は、上記制御フローが、プログラムカ

6

ウンタPCに連続するアドレスに進むのではなく、分岐先アドレスの方向に進む可能性があるため、データバス部20のセレクトSel. 9に、条件Jump Op.

（M）信号を送出し、セレクトSel. 9が、Branch用プログラムカウンタPC1の出力をJump Address（M）に切り替える。一方、Pre Decoder（M）111がブリデコードし、分岐条件判定回路131が分岐命令ではないと判定した場合は、ここでは何も行われず、これ以降は、通常のプロセッサと同様に動作する。

【0028】Address Unit（M）115は、Pre Decoder（M）111から入力した即値113と、Register File22から入力されたRegister値とに基づいて、アドレス計算に関する情報に従ってアドレス計算を行い、Jump Address（M）を、データバス部20のセレクトSel. 3とSel. 9とにフォワードする。

【0029】次に、分岐制御フロー処理部12について説明する。

【0030】分岐制御フロー処理部12は、Branch用プログラムカウンタBranch PC1が示すアドレスに応じて読み出された命令コードについて、主に、分岐命令を検出し、分岐先アドレスを計算する部分である。上記のように、非分岐制御フロー処理部11が条件分岐命令を検出した場合、その分岐先アドレスがフォワードされ、プログラムカウンタBranch PC1の出力と置き換えられる。そして、プログラムカウンタBranch PC1には、この分岐先アドレスがインクリメントされたアドレスが新たにセットされる。したがって、プログラムカウンタBranch PC1から出力されるアドレスは、分岐成立の場合の制御フローのアドレスであるとなすことができる。

【0031】リードポートP2を介して、プログラムカウンタBranch PC1から出力されたアドレスに応じて読み出された命令コードは、Branch IR（インストラクションレジスタ）16に入力される。分岐制御フロー処理部12のPre Decoder（B）121は、直ちに、Branch IR16で保持された命令コードについて、制御フローを変化させる分岐命令（無条件ジャンプ/Call/Return）の検出と分岐情報の抽出とを行う。なお、Branch IR16には、条件分岐命令が入力されないため、この命令に関する処理を行う必要はない。

【0032】つまり、プロセッサ1において、条件分岐命令は、その直前に1つ以上のテスト命令を必ず伴うものとするので、条件分岐命令が連続して発生することがない。なお、上記テスト命令は、CCRの状態を設定する演算命令であり、上記CCRは、コンディションコードレジスタである。

【0033】また、プロセッサ1において、IFステ

(5)

特開平9-282163

7

ジでMain IR15とBranch IR16とにフェッチされた2方向の制御フローは、次のクロックでどちらか一方に決定され、新たにフェッチされる命令コードは、Main IR15にフォワードされるので、Pre Decoder (B) 121での処理が有効なのは、分岐命令の直後に分岐先アドレスに応じて読み出される1命令のみだからである。

【0034】Pre Decoder (B) 121は、Pre Decoder (M) 111における上記処理と同様の処理を行い、この場合、信号の出力先が異なるだけである。すなわち、即値やアドレス計算に関する情報をAddress Unit (B) 122に送出し、Register Select (B) 信号をRegister File 22へ送出する。

【0035】また、CallまたはReturn命令を検出した場合は、Call Op. (B) 信号とRet Op. (B) 信号とを、Push/Pop制御回路部14に送出し、無条件ジャンプ命令(Return命令を含む)が検出された場合は、無条件Jump Op. (B) 信号をデータバス部20のセレクトSel. 7に

送出し、セレクトSel. 7は、プログラムカウンタBranch PC1の出力をJump Address (B) に切り替える。

【0036】Address Unit (B) 122は、Address Unit (M) 115における上記処理と同様に処理する。指定されたアドレス計算を行い、Jump Address (B) 信号をデータバス部20のセレクトSel. 8にフォワードする。

【0037】次に、分岐条件判定回路部13について説明する。

【0038】分岐条件判定回路部13は、条件分岐命令によって2方向に別れた制御フローから正しい方向を選択し、上記の命令フェッチ動作を、プログラムカウンタPCとMain IR15との組み合わせを用いた通常のバスに戻す制御を行う部分である。

【0039】分岐条件判定回路部13は、IFステージでPre Decoder (M) 111が抽出した分岐条件112と先行するテスト命令とを、EXステージで実行することによって、ALUから直接フォワードされるCCR状態信号を入力とし、これらの情報から分岐の成立/不成立を判定する回路である。つまり、たとえば、フラグ=1ならば分岐し、フラグ=1ならば分岐しないという分岐条件112をPre Decoder (M) 111が抽出し、先行するテスト命令についてALUがテストした結果、フラグ=1であった場合、フラグ=1というCCR状態信号をALUが出力し、分岐条件判定回路131が、上記分岐条件と上記CCR状態信号とに基づいて、分岐すべしとする信号が出力される(分岐条件判定回路131が出力するJump Enable信号が真になる)。

8

【0040】そして、分岐成立が判定された場合、分岐条件判定回路部13は、Jump Enable信号を、セレクトSel. 1、Sel. 2と、Push/Pop制御回路部14の論理積素子AND1、AND2、AND3、AND4と、データバス部20のセレクトSel. 5、Sel. 6とに送出する。

【0041】セレクトSel. 1は、分岐条件判定回路部13が出力するJump Enable信号が偽である場合、Main IR15に保持されている命令コードをIDステージのデコーダに送出し、一方、分岐条件判定回路部13が出力するJump Enable信号が真である場合、Branch IR16に保持されている命令コードをIDステージのデコーダに送出することによって、分岐の有無に応じて決定された命令コード(正しい制御フロー上の命令コード)がIDステージに投入されることを保証する。

【0042】セレクトSel. 2は、分岐条件判定回路部13が出力するJump Enable信号が偽である場合、リードポートP1から読み出された命令コードをIFステージのMain IR15に送出し、Jump Enable信号が真である場合、リードポートP2から読み出された命令コードを、IFステージのMain IR15に送出することによって、条件分岐による2方向の命令フェッチ状態が解消された後における分岐の有無に応じて決定された命令コード(正しい制御フロー上の命令コード)が、Main IR15に投入されることを保証する。

【0043】データバス部20のセレクトSel. 5、Sel. 6に、Jump Enable信号を送出するのは、分岐の有無に応じて決定された命令コードのアドレスをプログラムカウンタPCにセットするためである。

【0044】次に、Push/Pop制御回路部14について説明する。

【0045】Push/Pop制御回路部14は、Call命令によるStack 212のPush操作またはReturn命令によるPop操作の有効/無効の判定を行う部分であり、論理積素子AND1、AND2、AND3、AND4と、論理和素子OR1、OR2とを有するものである。

【0046】素子AND1は、Main IR15に保持されているCall命令の有効性を示す信号を出力するものであり、Pre Decoder (M) 111からのCall Op. (M) 信号と、分岐条件判定回路部13からのJump Enable信号の反転信号とを論理積するものである。つまり、Main IR15に保持されている命令コードが、分岐の有無に応じて決定された命令コードであり、しかも、この命令がCall命令であるときに、素子AND1の出力は真となる。

【0047】素子AND2は、Main IR15に保

(6)

特開平9-282163

9

10

持されているReturn命令の有効性を示す信号を出力するものであり、素子AND3は、Branch IR16に保持されているCall命令の有効性を示す信号を出力するものであり、素子AND4は、Branch IR16に保持されているReturn命令の有効性を示す信号を出力するものである。

【0048】なお、上記実施例ではデータバス部20にJump Enable信号を供給し、Stack212へのアドレスデータ入力をセクタSel. 6で制御することによって、2つの制御フローの選択を行っている。素子AND1の出力と素子AND3の出力とを素子OR1でまとめ、これをPush Enable信号としてデータバス部20のStack212に送出している。同様に、素子AND2の出力と素子AND4の出力とを素子OR2でまとめ、これをPop Enable信号として、データバス部20のセクタSel. 3、Sel. 8とStack212に送出している。

【0049】次に、データバス部20のプログラムカウンタ制御部21における各セクタの動作とその役割について説明する。

【0050】プログラムカウンタ制御部21は、プログラムカウンタPCと、Branch用プログラムカウンタBranch PC1と、セクタSel. 3、Sel. 4、Sel. 5、Sel. 6、Sel. 7、Sel. 8、Sel. 9と、Stack212とを有するものである。

【0051】セクタSel. 3は、Pop Enable信号が偽である場合に、Jump Address (M)を送出し、Pop Enable信号が真である場合に、StackからPopされたアドレスをセクタSel. 4に送出するものである。また、セクタSel. 4は、無条件Jump Op. (M)が偽である場合に、プログラムカウンタPCが出力するアドレスを出力し、無条件Jump Op. (M)が真である場合に、セクタSel. 3から入力されるアドレスをセクタSel. 5とリードポートP1に出力するものである。

【0052】すなわち、このセクタ部分は、非分岐制御フロー処理部11で無条件に制御フローが変更される命令（無条件ジャンプ、Return等）が検出された場合に、プログラムカウンタPCからのアドレス出力を分岐先アドレスまたはStack212からポップされたアドレスに置き換える役割を持つ。

【0053】セクタSel. 8は、Pop Enable信号が偽である場合に、Jump Address (B)を送出し、Pop Enable信号が真である場合に、Stack212からPopされたアドレスをセクタSel. 7に送出するものである。

【0054】また、セクタSel. 7は、無条件Jump Op. (B)が偽である場合に、プログラムカウンタBranch PC1から出力されるアドレスを出力

し、無条件Jump Op. (B)が真である場合に、セクタSel. 8から入力されたアドレスをセクタSel. 5とSel. 9とに出力する。すなわち、このセクタSel. 7は、分岐制御フロー処理部12で無条件に制御フローが変更される命令（無条件ジャンプ、Return等）が検出された場合に、プログラムカウンタBranch PC1からのアドレス出力を、分岐先アドレスまたはStack212からポップされたアドレスに置き換える役割を持つ。

【0055】セクタSel. 9は、条件Jump Op. (M)信号が偽である場合に、セクタSel. 7から入力されるアドレスを出力し、条件Jump Op. (M)信号が真である場合に、Jump Address (M)をリードポートP2に出力するものである。リードポートP2による命令コードのアクセスが必要になるのは、非分岐制御フロー処理部11において条件分岐命令が検出されたために分岐先アドレスの命令コードを（Jump Address (M)で示されるアドレスで）アクセスする場合と、Branch IR16にフェッチされた命令コードの次の命令コードを、セクタSel. 7から出力されるアドレスでアクセスする場合との、高々2クロック時間である。そしてこれらは決して重複しない。したがって、このセクタSel. 9には、次の①、②の2つの役割が与えられている。

① 非分岐制御フロー処理部11において条件分岐命令が検出された場合に、分岐先の命令コードをBranch IR16にフェッチするので、リードポートP2に対して分岐先アドレスを出力するとともに、次クロックの命令フェッチに備えるため、このアドレスをインクリメント213してプログラムカウンタBranch PC1にフィードバックする。

② 分岐先の命令コードがBranch IR16にフェッチされた後、次の命令コードをフェッチするため、リードポートP2に対してプログラムカウンタBranch PC1から出力されるアドレスまたは、無条件ジャンプ命令等によって、これに置き換わるアドレスを出力する。

【0056】セクタSel. 5は、Jump Enable信号が偽である場合に、セクタSel. 4から入力されるアドレスを送出し、Jump Enable信号が真である場合に、セクタSel. 7から入力されるアドレスを送出するものである。すなわち、このセクタSel. 5は、条件分岐命令の分岐成立／不成立判定によって決定される正しい方の制御フローのアドレスを選択し、インクリメント211してプログラムカウンタPCに設定する役割を持つ。

【0057】セクタSel. 6は、Jump Enable信号が偽である場合に、プログラムカウンタPCから出力されるアドレスを送出し、Jump Enable信号が真である場合に、プログラムカウンタBra

(7)

特開平9-282163

11

Branch PC1から出力されるアドレスをStack 212に送出する。すなわち、このセレクトSel. 6は、条件分岐命令の分岐成立／不成立判定によって決定される分岐の有無に応じて決定された命令コードのアドレスを選択し、Stack 212に送出する役割を持つ。

【0058】Stack 212は、Push Enable信号が真である場合に、入力されるデータをスタックにプッシュし、また、Pop Enable信号が真である場合に、スタックからデータをポップし出力するものである。

【0059】上記実施例によれば、分岐命令専用のブリデコード回路111がIFステージに組み込まれることによって、IFステージで読み出した命令コードに基づいて、同一クロック内で分岐またはジャンプ命令を検出し、その分岐情報を取り出すことが可能になる。さらに、取り出した分岐情報に基づいて、アドレス計算回路を用い、実際の分岐先アドレスを計算するようにしたので、プロセッサ1の様々なアドレッシングモードに対応することができる。

【0060】また、次のクロックのIFステージで、従来の命令フェッチ動作と分岐先命令のフェッチ動作との両方の動作を行う。ここで、「従来の命令フェッチ動作」とは、プログラムカウンタPCをインクリメントした値をアドレスとし、フェッチした命令コードをインストラクションレジスタにセットする処理のことであり、直前の命令が分岐／ジャンプ命令であった場合は、分岐不成立の制御フローに相当する動作である。また、「分岐先命令のフェッチ動作」とは、アドレス計算回路によって計算された値をアドレスとし、フェッチした命令コードをBranch IR16にセットする処理のことであり、直前の命令が分岐／ジャンプ命令であった場合は、分岐成立の制御フローに相当する動作である。したがって、条件分岐または無条件ジャンプ命令がフェッチされた次のクロックのIFステージで、分岐成立の場合の命令コードと不成立の場合の命令コードとの両方の命令コードをフェッチすることが可能になる。

【0061】分岐の成立／不成立は、たとえばCCR（コンディションコードレジスタ）の状態情報と、分岐命令で指定される分岐条件とから判定される。そして、テスト命令（CCRの状態を設定する演算命令）は、分岐命令よりも先行してパイプラインに投入される。したがって、たとえばテスト命令がEXステージ、分岐命令がIDステージ、分岐成立と分岐不成立との2つの命令がIFステージにある場合、次のクロックに移行する前までには、上記テスト命令によってセットされるCCRの状態と、前のクロックで、ブリデコード回路111によって分岐命令から抽出された分岐条件とから、分岐の有無が判定可能である。プロセッサ1は、この判定結果に応じて通常のインストラクションレジスタの出力と、

12

Branch IR16の出力との方の一方を選択し、命令デコードステージに送り込むので、常に、正しい命令コードIDステージに投入することが可能であり、矛盾の無い正常な制御フローを確保することができる。

【0062】つまり、上記実施例は、n段パイプライン化された各種プログラム処理方式のプロセッサにおいて、インストラクションフェッチステージでフェッチした分岐命令から分岐先アドレスを取り出し、分岐不成立の場合の制御フローと分岐成立の場合の制御フローとの両方の制御フローに対応する命令コードを、次のクロックのインストラクションステージでフェッチし、先行するテスト命令をALUが実行した結果と分岐条件とに基づいて、分岐の有無に応じて決定された命令コードを選択し、この選択された命令コードをインストラクションデコードステージに投入するダブルインストラクションフェッチプロセッサである。

【0063】また、上記実施例において、Pre Decoder (M) 111は、分岐情報を含む命令コードを命令フェッチステージで検出し、この検出された命令コードの中から、上記分岐情報を取り出すブリデコード回路の例である。Address Unit (M) 115は、上記ブリデコード回路で取り出された上記分岐情報に基づいて、分岐先アドレスを決定するアドレス計算回路の例である。Branch IR16は、上記アドレス計算回路で決定されたアドレスにアクセスし、分岐先の命令コードをフェッチし、このフェッチされた分岐先の命令コードを格納するブランチ用インストラクションレジスタの例である。セレクトSel. 1は、分岐成立または分岐不成立を決める分岐条件についての判定結果に応じて、通常のインストラクションレジスタの出力と上記ブランチ用インストラクションレジスタの出力とのうちの一方の出力を選択し、この選択された出力を命令デコードステージに送り込むインストラクションレジスタ選択手段の例である。

【0064】なお、上記実施例を構成するブリデコード回路111、アドレス計算回路、分岐条件判定回路部13は、従来のプロセッサにおけるインストラクションデコードステージのデコーダ部分に集中して置かれているものであり、また、これら以外には、少量のレジスタ、配線要素、基本的なゲート素子、セレクト回路等で上記実施例を実現できるので、上記実施例を実現するために必要な新規のハードウェアリソースは極めて少ない。

【0065】

【発明の効果】本発明によれば、インストラクションフェッチステージでフェッチした分岐命令から分岐先アドレスを取り出し、分岐不成立の場合の命令コードと分岐成立の場合の命令コードとの両方の命令コードで、次のクロックのインストラクションステージでフェッチし、先行するテスト命令の実行結果と分岐条件とに基づいて



(8)

特開平9-282163

13

て、分岐の有無に応じて決定された命令コードを選択し、インストラクションデコードステージに投入するので、パイプライン化されたプロセッサにおいて、小規模のハードウェアによって、常に正常な制御フローを保證することができ、制御ハザードによるパイプラインストールが発生しないという効果を奏する。

【図面の簡単な説明】

【図1】本発明の一実施例である書積プログラム処理方式のダブルインストラクションフェッチプロセッサ1を示すブロック図である。

【図2】ダブルインストラクションフェッチプロセッサ1における制御部10を、プロセッサ内部キャッシュ30とともに示す図である。

【図3】ダブルインストラクションフェッチプロセッサ1におけるデータバス部20を、プロセッサ内部キャッシュ30とともに示す図である。

14

【符号の説明】

1…ダブルインストラクションフェッチプロセッサ、  
10…制御部、  
11…非分岐制御フロー処理部、  
111…Pre Decoder (M)、  
115…Address Unit (M)、  
12…分岐制御フロー処理部、  
121…Pre Decoder (B)、  
122…Address Unit (B)、  
20…データバス部、  
21…プログラムカウンタ制御部、  
212…Stack、  
Sel. 1～Sel. 9…セクタ、  
22…Register File、  
30…プロセッサ内部キャッシュ。

10

1: ダブルインストラクションフェッチプロセッサ



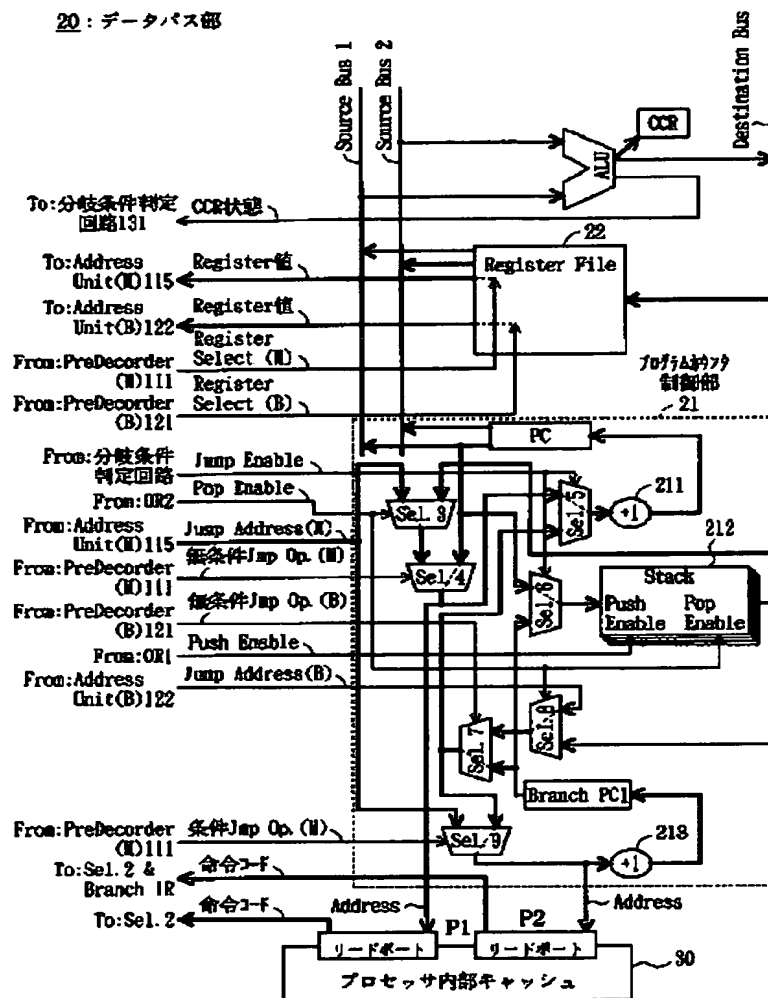


(11)

特開平9-282163

【図3】

20: データバス部



K3780

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**